Part 1

2/22/2016

Work in groups of 2-3 students:

* No "teams" of 1 person allowed, no exceptions. This part of the assignment is strictly a team project.
* Sign up yourselves into one of the **Assignment4\_Design** [Canvas groups created for Assignment 4](https://canvas.uw.edu/courses/1032130/groups#tab-37674)

Design Requirements

Your design should document *all* the work that needs to be done to complete the assignment. It should be a complete and clear description of how the program is organized. The more time you spend on your design, the less you will spend coding, debugging, and modifying your code.

As you work through your design, think: who is your "customer"? what would such a customer would like to see?  Remember, design should be goal-oriented, and goals are driven by the end-users of our product.

Your design should include (*at least*) the following components **in this order**:

* **Overview**:  This is a description of the design and how the pieces fit together (the interaction between the classes). Include a description of main (List the objects that you have. Main should be short.)
* **Problem Description:** do not copy-paste the problem description from this assignment, instead, state what is your interpretation of the problem to solve
* **Assumptions and design considerations**: describe any relevant assumptions and or considerations that will impact your design and / or implementation
* **High-level algorithms**(s)
* **Class diagram**:  This is a UML diagram showing class relationships, including inheritance and composition. You can use Visio or **neatly** draw this by hand.
* **Class descriptions**:  For each class in the design, describe the data and methods as part of a documented C++ header file (**Exception**: you do not need to include h files of classes you will not implement, of the extensions beyond the assignment specifications, but you must include a description of those classes). The task that each function performs and the purpose of each data member should be clearly described. High-level pseudo code should be included for the most important methods (for example, those that control the flow of the program). Not all parameters need to be included for methods. Please order the files properly, i.e., put the most important classes first, put parent classes before children classes.

You are expected to put considerable effort into the design (comparable to or above a lab implementation).  Sloppy incomplete designs will not receive a good score.

In real-world applications, the purpose of design is more than often to provide guidance for someone who is not familiar with the project, e.g.,  you are writing a design that someone else will need to implement and/or extend. Document your design the same way you would expect someone else to document for you.

Do NOT implement any code at this point.

Some questions you might want to ask yourself:

* Can your design be extended beyond the specifications provided in here?
* Could you easily add new videos or DVDs to your design?
* Can you easily add other categories of videos or DVDs?
* Could you easily add new categories of media to your design, for example, music?
* Could you expand to check out other kinds of items, for example VCRs or DVD players?
* Could you easily add new operations to your design?
* Could you incorporate time, for example, a due date for borrowed items?
* Could you easily add an additional store, or handle a chain of stores?

**Your design can go beyond the scope of these specifications (you won’t need to implement extensions).** Thinking of possible extensions in advance often improves the design.

***Bring your design on paper to class!!!***

* If you don't bring the "hard copy" to class you will receive no points for this part of the assignment
* We will have in-class design reviews and then your write-up will be turned in.
* **You are also required to submit an electronic copy of your design in Canvas.**